Another look at weak feedback polynomials in the nonlinear combiner

Hell, Martin; Brynielsson, Lennart

2009

Another Look at Weak Feedback Polynomials in the Nonlinear Combiner

Martin Hell
Department of Electrical and Information Technology
Lund University, Sweden
E-mail: martin@eit.lth.se

Lennart Brynielsson
MUST/TSA
Swedish Armed Forces
107 86 Stockholm, Sweden

***Abstract*—Feedback polynomials with low degree multiples of low weight should be avoided in linear feedback shift registers when used in nonlinear combiners. We consider another class of weak feedback polynomials, namely the class when taps are located in small groups. This class was introduced in 2004 demonstrating that the resulting distinguishing attack can sometimes be better than the one using low weight multiples. In this paper we take another look at these polynomials and give further insight to the theory behind the attack complexity. Using the Walsh transform we show an easy way to determine the attack complexity given a polynomial. Further, we show that the size of the vectors should sometimes be larger than previously known. We also give a simple relation showing when the new attack will outperform the simple attack based on low weight multiples.**

I. Introduction

Symmetric key encryption primitives are either implemented as stream ciphers or block ciphers. Whereas a block cipher takes a b-bit input string and outputs a b-bit output using a key dependent permutation on b-bit strings, a stream cipher traditionally operates on single bits. Each output bit depends on an internal state and an output function and the internal state is updated for each output. To take advantage of modern processors it is also common to design stream ciphers operating on words instead of bits. While block ciphers are practically always based on either the Feistel structure or the SP-network, see [1], there are many ways to design stream ciphers. One of the most common building blocks is the Linear Feedback Shift Register (LFSR). It is used in some of the most common stream ciphers, e.g., A5/1 in GSM and E_0 in Bluetooth. Two important design ideas based on LFSRs are the nonlinear combiner and the nonlinear filter generator. This paper will focus on the nonlinear combiner. It is well known that LFSRs with feedback polynomials of low weight should be avoided in stream ciphers. An attack exploiting low weight feedback polynomials is the fast correlation attack, proposed by Meier and Staffelbach [2]. Such polynomials provide a low weight parity check equation which can be exploited together with a bias in the output function. Because of this attack, low weight feedback polynomials can be considered weak in the context of stream cipher design. Much research has been put into the fast correlation attack and several improvements have been found, resulting in the fact that this attack is one of the most important cryptanalytic attacks on stream ciphers. In 2004, Englund, Hell and Johansson introduced a new class of weak feedback polynomials [3]. These polynomials were not necessarily of low weight, but instead had their feedback taps located in groups. The groups can be far apart but the distance between the first and last tap in each group should be small. In this paper we take another look at these polynomials and, using the Walsh transform, we show better and more efficient ways of how to compute the complexity of the resulting attack. Using our analysis we show that the length of the vectors used in the attack should sometimes be larger than used in [3]. We also use our analysis to give a simple relation stating when these weak feedback polynomials can be used to mount a more efficient attack than the attack first proposed by Meier and Staffelbach. The paper is outlined as follows. In Section II we give some background theory. In Section III we discuss the previous work done in [3], give the open problems from that paper and motivate why further analysis is needed. Then we give our analysis of the problem in Section IV. In Section V we show an example of a polynomial giving a very efficient attack using a relatively large vector length. In Section VI we give an easy way to check whether the new attack will outperform the basic attack by Meier and Staffelbach when considering low weight multiples of the feedback polynomial. Finally, the paper is concluded in Section VII.

II. Preliminaries

Consider the fast correlation attack [2]. The ideas behind the fast correlation attack, originally given as a key recovery attack on a nonlinear combiner, can easily be turned into a distinguishing attack on the same nonlinear combiner. The nonlinear combiner uses a set of T LFSRs, preferably with primitive feedback polynomials, and a nonlinear Boolean output function. We denote the ith LFSR by R_i and its size by L_i. The output of R_i at time t is denoted $x_i(t)$.

The correlation attack relies on the fact that there is always a subset of the shift registers such that

$$\Pr(z(t) = x_{i_1}(t) \oplus x_{i_2}(t) \oplus \ldots \oplus x_{i_b}(t)) = \frac{1}{2}(1+\varepsilon) \quad (1)$$

where $z(t)$ is the output of the Boolean function, or the keystream. The largest value of b such that $\varepsilon = 0$ for all choices of $i_1, \ldots, i_b$ is called the resiliency m of the Boolean function, assuming that the function is balanced. Relating

this to information theory, we can also say that the mutual information between the output and any subset of b inputs to the Boolean function is zero if $b \leq m$. A well known result [4] is the relationship $m + d < n$ where n is the number of variables, d the algebraic degree and m the resiliency of a balanced Boolean function. The only exception is the parity check function with $d = 1$ and $m = n - 1$. This result shows that a subset of inputs will *always* give us information about the output in a nonlinear combiner.

The characteristic polynomial $f(x)$ of the sequence $s(t) = x_{i_1}(t) \oplus \ldots \oplus x_{i_b}(t)$ is given by

$$\begin{aligned} f(x) &= lcm(f_{i_1}(x), f_{i_2}(x), \ldots, f_{i_b}(x)), & (2) \\ &= c_0 + c_1 x + c_2 x^2 + \ldots c_L x^L, & (3) \end{aligned}$$

where f_{i_j} is the characteristic polynomial of the sequence generated by R_{i_j} and *lcm* is the least common multiple. If the polynomials are primitive, (2) is reduced to the product of the involved polynomials and the degree of $f(x)$, equivalent to the size of the corresponding LFSR, is $L = \sum_{j=1}^{b} L_{i_j}$. The fast correlation attack takes advantage of the fact that we can find a multiple of $f(x)$ which is of low weight, giving us a parity check equation of low weight that holds with probability $\neq 0.5$. In the remainder of this paper, we will assume that the set of LFSRs has been replaced by one LFSR according to (2) and for clarity of presentation we will from now on use the notation s_t to denote the value of sequence s at time instance t.

In this paper we will use the Walsh transform of a probability distribution in our analysis. Let ω and x be vectors of the same length and let $\omega \cdot x$ be the scalar (or dot) product of the two vectors taken modulo 2. Then the ωth order Walsh function of x is defined as

$$h_\omega(x) = (-1)^{\omega \cdot x}. \quad (4)$$

Denote the probability distribution function of the random variable X by $P_X(x)$. Then the Walsh transform of $P_X(x)$ is defined as the expectation of $h_\omega(X)$,

$$W_{P_X}(\omega) = E[h_\omega(X)] = \sum_x h_\omega(x) P_X(x). \quad (5)$$

Combining (4) and (5) we get

$$W_{P_X}(\omega) = \Pr(\omega \cdot X = 0) - \Pr(\omega \cdot X = 1), \quad (6)$$

a relation which will turn out to be helpful in Section IV. Note that from (4) it follows that

$$h_\omega(x) = 1 - 2(\omega \cdot x). \quad (7)$$

For a more thorough treatment of the Walsh transform and its applicability in statistics we refer to [5].

Hypothesis testing is a central component in a distinguishing attack. It can be used to decide if an observed collection of samples are drawn from a biased distribution, here called the cipher distribution P_C, or from a uniform distribution P_0. For an overview of hypothesis testing we refer to [6]. Its application to cryptanalysis is treated in e.g., [7], [8]. Here we only give the prerequisites necessary for the presentation of our results. The Kullback-Leibler distance, also known as divergence or relative entropy, between two distributions is defined as

$$D(P_C \| P_0) = \sum_x \Pr_{P_C}(x) \log \frac{\Pr_{P_C}(x)}{\Pr_{P_0}(x)}. \quad (8)$$

The number of samples needed in the hypothesis test is in the order of $O(1/D(P_C \| P_0))$.

III. Previous work

In this section we review the results given in [3]. Consider Eq. (1). Replacing the LFSRs used in the stream cipher by $f(x)$ as given in (2) we can write $z_t = s_t \oplus e_t$, where e_t is the noise introduced by the approximation referred to in (1). We see that

$$\Pr(e_t = 0) = \frac{1}{2}(1 + \varepsilon), \quad (9)$$

assuming that the Boolean output function is balanced. The linear polynomial $f(x)$ defines the recurrence

$$\bigoplus_{i=0}^{L} c_i s_{t+i} = 0, \quad t \geq 0. \quad (10)$$

Thus, we can write

$$\bigoplus_{i=0}^{L} c_i z_{t+i} = \bigoplus_{i=0}^{L} c_i s_{t+i} \oplus \bigoplus_{i=0}^{L} c_i e_{t+i} = \bigoplus_{i=0}^{L} c_i e_{t+i}. \quad (11)$$

We assume that all noise variables e_i are independent, an assumption that will be implicit throughout the paper. Then, according to the Piling-up Lemma [9] we know that

$$\Pr\left(\bigoplus_{i=0}^{L} c_i z_{t+i} = 0\right) = \Pr\left(\bigoplus_{i=0}^{L} c_i e_{t+i} = 0\right) = \frac{1}{2}(1 + \varepsilon^w), \quad (12)$$

where w is the Hamming weight of $(c_0, c_1, \ldots, c_L)$. This results in a distinguishing attack requiring $O(L + 1/\varepsilon^{2w})$ keystream bits. L is the distance between the first and last keystream bit in each sample and $1/\varepsilon^2$ is a common rule of thumb widely used in cryptanalysis to approximate the number of samples needed to detect the bias ε in (9). The complexity of the attack is highly dependent on the weight w of $f(x)$ and thus, it is usually favourable to consider low weight multiples of $f(x)$ instead. The degree L of a multiple is of course larger than the degree of the original $f(x)$, a property which constitutes a tradeoff when considering the number of required keystream bits, i.e., smaller weight w is traded for larger degree L. We will refer to this attack as the *basic attack*. The idea proposed in [3] was to generalize this attack and consider the case when $f(x)$ can be written as

$$f(x) = g_0(x) + x^{M_1} g_1(x) + x^{M_2} g_2(x) + \ldots + x^{M_\ell} g_\ell(x), \quad (13)$$

where $g_i(x)$ are polynomials of small degree ($\leq k$) and $M_1 < M_2 < \ldots < M_\ell$. It is also possible to consider multiples of this form. The polynomial (13) corresponds to an LFSR with taps placed in groups. Each group has taps at most k shift

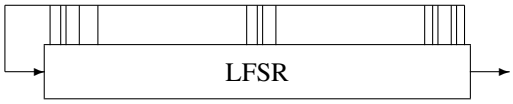

Fig. 1. The polynomial $f(x) = g_0(x)+x^{M_1}g_1(x)+x^{M_2}g_2(x)$ corresponds to an LFSR with taps concentrated to three groups.

register cells apart and groups are located far away from each other, see Fig. 1.

Now, introduce the variable Q_i,

$$Q_i = \underline{g_0} \cdot e[i, i+k] \oplus \ldots \oplus \underline{g_\ell} \cdot e[M_\ell + i, M_\ell + i + k] \quad (14)$$

where $e[i,j] = (e_i, \ldots, e_j)^T$ and $\underline{g_i} = (g_{i,0}, g_{i,1}, \ldots, g_{i,k})$, $g_{i,j}$ is the jth coefficient in the polynomial $g_i(x)$. The observation is now that even though consecutive noise variables e_i are independent, variables Q_i close together will be dependent. The reason is that the same noise variable will be used in Q_i's close together. Hence, we consider the noise vector of length N given by

$$E_i = (Q_{N\cdot i}, Q_{N\cdot i+1}, \ldots, Q_{N(i+1)-1}). \quad (15)$$

E_i can also be written as

$$E_i = \bigoplus_{j=0}^{\ell} G_j \cdot (e_{N\cdot i+M_j}, \ldots, e_{N(i+1)+M_j+k-1})^T, \quad (16)$$

where $\oplus$ denotes bitwise xor of binary vectors, $M_0 = 0$ and G_j is the size $N \times (N+k)$ matrix

$$G_j = \begin{pmatrix} g_{j,0} & g_{j,1} & \cdots & g_{j,k} & & \\ & g_{j,0} & \cdots & g_{j,k-1} & g_{j,k} & \\ & & & \vdots & & \\ & & g_{j,0} & g_{j,1} & \cdots & g_{j,k} \end{pmatrix}. \quad (17)$$

The efficiency of the distinguishing attack depends on the distribution of the vector E_i. We can note that since the different g_i are far apart their contribution to the total noise vector can be computed independently. In [3], different combinations of g_i's were tested and it was noted that for some combinations, the number of keystream bits needed in the distinguishing attack was significantly lower than in the basic binary attack. In particular, when comparing vectors of length N and $N+1$, for some values of N the attack was improved significantly, while for other values of N the attack did not improve at all. For some combinations of g_i's the attack even seemed to give the same performance as the basic attack. However, the authors did not manage to find the exact reason for this behaviour. Questions that remained to be answered were

- Which combinations of g_i's will give a more efficient attack?
- Which vector length is needed to get a significant improvement?

In Section IV we will provide very simple relations that can be used to answer these questions. We also show how to efficiently calculate the number of keystream bits needed in the attack.

IV. Analysis using the Walsh Transform

In order to answer the questions given above we will look at the Walsh transform of the probability distributions. The notation from the previous sections will be used. As mentioned before, when we compute the probability distribution of the vector E_i we consider the contribution from different g_i independently. Thus, for the moment, we look only at the size $N+k$ random variable vector $X = (x_0, x_1, \ldots, x_{N+k-1})^T$ where x_i are independent binary random variables corresponding to the noise introduced by the linear approximation of the Boolean output function. Thus, we have $\Pr(x_i = 0) = \frac{1}{2}(1+\varepsilon)$. In the attack, we look at the sequence $Q_i = \sum_{j=0}^{k} g_{\cdot,j} e_i$ and construct a vector of N consecutive bits of Q_i. We introduce the size N random variable $Y = (y_0, y_1, \ldots, y_{N-1})^T$ and write

$$Y = GX. \quad (18)$$

Our goal is then to find $D(Y\|P_0)$ which approximates the inverse of the number of samples needed by the distinguisher. A straight forward algorithm is to assign all possible 2^{N+k} values to the vector X, and then compute the resulting Y. This will give us $P_Y(y)$. Then, the distributions for all $\ell+1$ parts in (15) are combined and $D(Y\|P_0)$ can be computed.

Instead, consider

$$D(P_Y\|P_0) = \sum_y \Pr_Y(Y=y) \log \frac{\Pr_Y(Y=y)}{2^{-N}}. \quad (19)$$

If the distribution P_Y is close to the uniform distribution, we can write $\Pr_Y(Y=y) = 2^{-N} + \varepsilon_y$, where $|\varepsilon_y|$ is small. By using $\log_2 x = \ln x / \ln 2$, the Taylor expansion $\ln(1+x) \approx x - x^2/2$ and the fact that $\sum_y \varepsilon_y = 0$, we can write (19) as

$$D(P_Y\|P_0) \approx \frac{1}{2\ln 2} \sum_y \frac{(\Pr_Y(y) - 2^{-N})^2}{2^{-N}} \quad (20)$$

$$= \frac{1}{2\ln 2} \sum_{\omega \neq 0} W_Y(\omega)^2. \quad (21)$$

The last equality follows from Parseval's relation

$$\sum_y g(y)^2 = 2^{-N} \sum_\omega W_g(\omega)^2, \quad (22)$$

and the fact that the Walsh transform for the probability distribution $P_Y(y)$ is the same as for $P_Y(y) - 2^{-N}$ except when $\omega = 0$. Then we instead have $W_{P_Y}(0) = 1$ and $W_{P_Y - 2^{-N}}(0) = 0$. Thus, $\omega = 0$ is excluded in the right hand part of (21).

The approximation (21) is only useful if it turns out to be much easier to compute W_{P_Y} than to compute the probability distribution P_Y. Next, we show that this is indeed the case. Recall that we define $\Pr(x_i = 0) = \frac{1}{2}(1+\varepsilon)$ and from this it follows that the xor sum of w independent x_i has probability $\Pr(x_0 \oplus x_1 \oplus \ldots \oplus x_{w-1} = 0) = \frac{1}{2}(1+\varepsilon^w)$. Thus, we have

$$\Pr\left(\bigoplus_{j=0}^{w-1} x_i = 0\right) - \Pr\left(\bigoplus_{j=0}^{w-1} x_i = 1\right) = \varepsilon^w. \quad (23)$$

We can relate this to the Walsh transform of the probability distribution of the random variable $X = (x_0, x_1, \ldots, x_{N+k-1})^T$ which is given by

$$W_{P_X}(\omega) = \Pr(\omega \cdot X = 0) - \Pr(\omega \cdot X = 1) = \varepsilon^{\|\omega\|_1}, \quad (24)$$

where $\|\omega\|_1$ is the 1-norm, or Hamming weight, of ω. If we instead look at the Walsh transform of the probability distribution P_Y we get

$$\begin{aligned} W_{P_Y}(\omega) &= \Pr(\omega Y = 0) - \Pr(\omega Y = 1) && (25)\\ &= \Pr(\omega G X = 0) - \Pr(\omega G X = 1) && (26)\\ &= \varepsilon^{\|\omega G\|_1}. && (27) \end{aligned}$$

The relative entropy can now be written as

$$D(P_Y \| P_0) \approx \frac{1}{2 \ln 2} \sum_{\omega \neq 0} \varepsilon^{2\|\omega G\|_1}. \quad (28)$$

If ε is small, (28) is dominated by the term for which we have the smallest $\|\omega G\|_1$ and we can write

$$D(P_Y \| P_0) \approx \frac{1}{2 \ln 2} \varepsilon^{2 \cdot \min_\omega(\|\omega G\|_1)}. \quad (29)$$

Since the number of samples needed in the distinguisher is given by $O(1/D(P_Y \| P_0))$, (29) can be immediately used to determine this number, as well as to determine which vector length results in a significant improvement of the attack.

Up to this point we have only considered one polynomial g_0 in (13). Now we extend the analysis to include ℓ polynomials $g_0, g_1, \ldots, g_\ell$, a more practical situation. We show that using the Walsh transform, this extension is very simple compared to the case when the probability distributions are combined. The random variable vector Y will be given by

$$\begin{aligned} Y &= Y_0 \oplus Y_1 \oplus \ldots \oplus Y_\ell, && (30)\\ &= G_0 X_0 \oplus G_1 X_1 \oplus \ldots \oplus G_\ell X_\ell, && (31) \end{aligned}$$

where $\oplus$ is bitwise xor of vectors and $X_0, \ldots, X_\ell$ are independent. Then we can write

$$\begin{aligned} W_{P_Y}(\omega) &= \Pr(\omega Y = 0) - \Pr(\omega Y = 1) && (32)\\ &= \Pr(\omega G_0 X_0 \oplus \ldots \oplus \omega G_\ell X_\ell = 0) && (33)\\ &\quad - \Pr(\omega G_0 X_0 \oplus \ldots \oplus \omega G_\ell X_\ell = 1) && (34)\\ &= \varepsilon^{\|\omega G_0\|_1 + \|\omega G_1\|_1 + \ldots + \|\omega G_\ell\|_1}. && (35) \end{aligned}$$

This can be seen as a generalization of (27). Similarly, a generalization of (29) is given by

$$D(P_Y \| P_0) \approx \frac{1}{2 \ln 2} \varepsilon^{2 \cdot \min_\omega\left(\sum_{i=0}^{\ell} \|\omega G_i\|_1\right)}. \quad (36)$$

Thus, the complexity of the attack depends on the smallest sum of the Hamming weights $\|\omega G_i\|_1$. Note that this is not the same as the sum of the smallest Hamming weights. This tells us that we can not use (29) independently for each polynomial since the minimum Hamming weight might stem from different ω for different g_i.

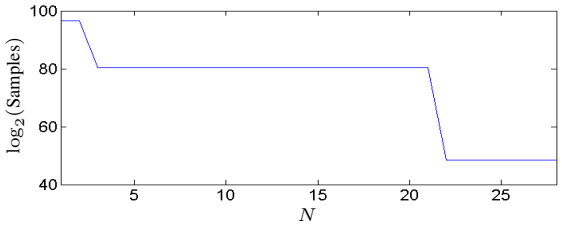

Fig. 2. The logarithm of the number of samples needed as a function of the vector length.

V. Simulations

Using the theory from Section IV we can find the number of samples needed in the distinguisher very easily. The theory also allows us to consider vectors of size much larger than given in [3]. In that paper results were given for vectors of length up to $N = 14$. We have simulated the efficiency of the distinguisher for all possible combinations of two polynomials of degree $k \leq 8$. Further, vectors of length up to $N = 25$ have been tested for all combinations. The simulations show that there are several cases in which the distinguisher is considerably better for vectors of larger length than $N = 14$. One example is $f(x) = g_0(x) + x^{M_1} g_1(x)$ with

$$\begin{aligned} g_0(x) &= 1 + x^2 + x^3 + x^4 + x^6 + x^7, && (37)\\ g_1(x) &= 1 + x + x^2 + x^5 + x^6 + x^8. && (38) \end{aligned}$$

In this case we get

$\min_\omega(\|\omega G_0\|_1 + \|\omega G_1\|_1)$	N
12	$1 \leq N \leq 2$
10	$3 \leq N \leq 21$
6	$22 \leq N \leq 28$

when considering vectors of length $N \leq 28$. As can be seen, if we would use vectors of length $3 \leq N \leq 21$ we would require in the order of ε^{-20} samples in order to distinguish the cipher distribution from a uniform distribution. If the vector size is $N \geq 22$ we would need at most in the order of ε^{-12} samples. The 2-logarithm of the number of samples needed as a function of the vector length, assuming $\varepsilon = 2^{-4}$, is given in Fig. 2. The basic attack, corresponding to $N = 1$, would require about 2^{96} samples to distinguish the cipher distribution from a uniform distribution. Using distributions of vectors, vector size at least 22, will allow the distinguishing attack to succeed with only about 2^{48} samples.

VI. Comparison with Basic Attack

While the example in Fig. 2 is a bit extreme and not very representative for a random combination of two polynomials of degree $k \leq 8$, it does show that this type of attack must be considered when constructing ciphers based on nonlinear combiners.

In practice, one would take advantage of the fact that a low weight multiple of the characteristic polynomial can be used

in the distinguisher. In [10], it was shown that the degree L' at which we expect to find a multiple of weight w is given by

$$L' \approx 2^{\frac{L}{w-1}}. \tag{39}$$

In [3], it was shown that if we are looking for a polynomial of the form (13) with t groups ($t = \ell+1$) of smaller polynomials g_i of degree at most k, we expect it to be of degree

$$L' \approx 2^{\frac{L-tk}{t-1}}. \tag{40}$$

From this we can conclude that it will practically never be advantageous to consider multiples of the form (13). The degree of a multiple with t groups is just slightly less than the degree of a multiple with weight w, when k is moderate. Thus, considering a low weight multiple will result in a better attack. Instead, we consider the case when the polynomial $f(x)$ turns out to be of the desired form (or perhaps a multiple of surprisingly low degree). We now compare this case with the basic attack when low weight multiples are used. The number of keystream bits T required in the basic attack is given by

$$T = 2^{\frac{L}{w-1}} + \varepsilon^{-2w}. \tag{41}$$

The smallest amount of keystream bits is achieved by choosing the w that minimizes (41). In practice this means choosing w such that $2^{\frac{L}{w-1}} \approx \varepsilon^{-2w}$. We use $\widehat{w}$ to denote this choice which depends on both L and ε. In Table I, we give $\widehat{w}$ for some values of L and ε. The number of keystream bits needed in the attack

TABLE I
VALUE OF $\widehat{w}$ FOR SOME CHOICES OF L AND ε.

	L				
	100	200	300	400	500
$\varepsilon = 2^{-3}$	5	6	8	9	10
$\varepsilon = 2^{-5}$	4	5	6	7	8
$\varepsilon = 2^{-7}$	3	4	5	6	7
$\varepsilon = 2^{-9}$	3	4	5	5	6

when $f(x)$ is of the form (13) is approximately $1/D(P_Y\|P_0)$ where $D(P_Y\|P_0)$ is given by (36). Ignoring small constants, this attack will be more efficient than the basic attack when

$$\varepsilon^{2\cdot\min_\omega\left(\sum_{i=0}^{\ell}\|\omega G_i\|_1\right)} > \varepsilon^{2w}. \tag{42}$$

$$\Rightarrow \quad \min_{\omega}\left(\sum_{i=0}^{\ell}\|\omega G_i\|_1\right) < \widehat{w}. \tag{43}$$

This can be seen as a rule of thumb and should be checked when designing stream ciphers based on the nonlinear combiner. We can note that for small ε and small L the basic attack is likely to still outperform our attack, whereas for large values of ε and L our attack is likely to be the most efficient.

VII. CONCLUSIONS

A distinguishing attack taking advantage of LFSR feedback taps located close together has been considered. The attack was first given in [3] and in this paper we have shown further results related to the attack. An easier and more efficient way to determine the efficiency of the attack has been given using the Walsh transform of probability distributions. Also, a simple relation showing if the attack will outperform the basic attack for a given feedback polynomial has been given.

Finally, we note that this analysis shows that there is an interesting relation to coding theory. The matrix G_i corresponds to the generator matrix of a cyclic code with generator polynomial $g_i(x)$. Thus, it might be possible to find further improvements by considering results from coding theory. This will be left as future work.

REFERENCES

[1] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997.

[2] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.

[3] H. Englund, M. Hell, and T. Johansson, "Correlation attacks using a new class of weak feedback polynomials." in *Fast Software Encryption 2004*, ser. Lecture Notes in Computer Science, B. Roy and W. Meier, Eds., vol. 3017. Springer-Verlag, 2004, pp. 127–142.

[4] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Transactions on Information Theory*, vol. 30, pp. 776–780, 1984.

[5] J. Pearl, "Application of walsh transform to statistical analysis," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 1, no. 2, pp. 111–119, 1971.

[6] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Wiley series in Telecommunication. Wiley, 1991.

[7] T. Baignères, P. Junod, and S. Vaudenay, "How far can we go beyond linear cryptanalysis?" in *Advances in Cryptology—ASIACRYPT 2004*, ser. Lecture Notes in Computer Science, vol. 3329. Springer-Verlag, 2004, pp. 432–450.

[8] M. Hell, T. Johansson, and L. Brynielsson, "An overview of distinguishing attacks on stream ciphers," *Cryptography and Communications*, 2008.

[9] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology—EUROCRYPT'93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed., vol. 765. Springer-Verlag, 1994, pp. 386–397.

[10] J. Golić, "Computation of low-weight parity-check polynomials," *Electronic Letters*, vol. 32, no. 21, pp. 1981–1982, October 1996.